REMARKS

Claims 1-22 are pending and rejected by the examiner. Claims 1, 8, 16 and 22 are independent claims.

The examiner rejected claim 19 under 35 U.S.C. §112, second paragraph for containing a spelling error.

Applicants amended claim 19 to correct the spelling error. No new matter was added.

The examiner uses the Intel IXP 1200 Hardware Reference Manual ("IXP 1200 Manual") to reject claims 1-21 as having been anticipated.

Independent claims 1, 8, 16 and 22 recite "inserting a segment of executable code into an unused section of a target microengine's microstore in response to a first context swap of one of a plurality of hardware-supported execution threads of a program executing in the target microengine," or similar language.

The IXP 1200 Manual neither describes nor suggests at least this quoted feature. On the contrary, the IXP 1200 Manual merely discloses the insertion of breakpoint instructions and not a segment of executable code. More specifically, the IXP 1200 Manual discloses:

> Breakpoints are supported by replacing instructions in the program with branch instructions to a breakpoint routine. (Section 4.17.3, lines 1-2)
>
> Breakpoints are inserted into a program by replacing the instruction where the breakpoint should occur with a branch instruction to a breakpoint routine. (Section 4.17.3, lines 13-14)

This is different from inserting a segment of executable code into an unused section of a target microengine's microstore. Among its advantages, the method works by causing a controlling processor to divert a normal program flow of another processor to execute a segment of code instructions that have been inserted into an unused section of the target's microstore. The segment of code will instruct the target processor to become disabled/paused the next time any of its contexts has been swapped out. The segment of code will then branch back to the program location prior to the diversion, allowing a normal flow.

Accordingly, claims 1, 8, 16 and 22 are not anticipated by the ISP 1200 Manual.

The examiner uses Xu to reject claims 1-15 as having been anticipated.

As described above, independent claims 1 and 8 recite "inserting a segment of executable code into an unused section of a target microengine's microstore in response to a first context swap of one of a plurality of hardware-supported execution threads of a program executing in the target microengine," or similar language.

As with the IXP 1200 Manual, Xu neither describes nor suggests at least this quoted feature. On the contrary, Xu merely discloses a patch to a base-trampoline. More specifically, Xu discloses:

> For each runtime instrumentation request, Paradyn patches a jump to a base-trampoline, and relocates the instructions that were overwritten. Each base-trampoline has pre- and post-instrumentation sections. (page 2, col. 2)

Here again, this is very different from inserting a segment of executable code into an unused section of a target microengine's microstore. In Xu, this is even more apparent in Figure 1 in which the application program discloses a jump that causes execution to leave the application and execute the base-trampoline and mini-trampolines.

Accordingly, claims 1 and 8 are not anticipated by Xu.

The examiner uses the Intel IXP 1200 Manual to reject claim 22 as having been obvious. The examiner also uses Xu to reject claim 22 as having been obvious.

Applicant disagrees. As was discussed above with reference to claims 1-21, the Intel IXP 1200 Manual and Xu do not teach at least the feature of inserting a segment of executable code into an unused section of the target microengine's microstore. Moreover, one skilled in this art would not be lead to provide inserting a segment of executable code into an unused section of the target microengine's microstore since both the Intel IXP 1200 Manual and Xu merely teach the insertion of a branch or jump instruction in the application, and not the insertion of a segment of executable code in the application, as claimed in claim 22. No one skilled in this art would associate an insertion of a breakpoint instruction or a jump instruction that diverts program flow outside of the application with the insertion of a segment of executable code within the

application. Accordingly, claim 22 is not rendered obvious by either the Intel IXP 1200 Manual or Xu.

The examiner uses Xu and Jacobson to reject claims 16-21 as having been obvious.

As discussed above, Xu fails to teach or suggest inserting a segment of executable code into an unused section of the target microengine's microstore. Jacobson fails to provide this deficiency in Xu as Jacobson merely teaches a basic architecture of a microengine. Inserting a segment of executable code into an unused section of the target microengine's microstore is totally absent in Jacobson and as such, claims 16-21 are not rendered obvious by Xu and Jacobson.

All of the dependent claims are patentable for at least the same reasons as the claims on which they depend.

The applicants' discussion of particular arguments of the Examiner should not be construed as a concession by the applicants with respect to any other positions of the Examiner. The applicants' assertion of arguments for patentability of certain claims should not be construed as suggesting that there are not also other good reasons why those or other claims are patentable.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: January 12, 2004

Kenneth F. Kozik
Reg. No. 36,572

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906